REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AIM 1101	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Routing Statistics for Unqueued Banyan Networks	5. TYPE OF REPORT & PERIOD COVERED: memorandum
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): Thomas F. Knight, Jr. and Patrick G. Sobalvarro	8. CONTRACT OR GRANT NUMBER(s): N00014-88-K-0825, N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Blvd., Arlington, VA 22209	12. REPORT DATE: September 1990 13. NUMBER OF PAGES: 20
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, VA 22217	15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

interconnection networks
prallel processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Banyan networks comprise a large class of networks that have been used for interconnection in large-scale multiprocessors and telephone switching systems. Regular variants of Banyan networks, such as delta and butterfly networks, have been used in multiprocessors such as the IBM RP3 and the BBN Butterfly. Analysis of the performance of Banyan networks has typically

(continued on back)

Block 20 continued:

focused on these regular variants. We present a methodology for performance analysis of unbuffered Banyan multistage interconnection networks. The methodology has two novel features: it allows analysis of networks where some inputs are more likely to be active than others, and allows analysis of Banyan networks of arbitrary topology.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1101 September, 1990

Routing Statistics for Unqueued Banyan Networks

Thomas F. Knight, Jr.
Patrick G. Sobalvarro

Abstract

Banyan networks comprise a large class of networks that have been used for interconnection in large-scale multiprocessors and telephone switching systems. Regular variants of Banyan networks, such as delta and butterfly networks, have been used in multiprocessors such as the IBM RP3 and the BBN Butterfly. Analysis of the performance of Banyan networks has typically focused on these regular variants. We present a methodology for performance analysis of unbuffered Banyan multistage interconnection networks. The methodology has two novel features: it allows analysis of networks where some inputs are more likely to be active than others, and allows analysis of Banyan networks of arbitrary topology.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contracts N00014-88-K-0825 and N00014-85-K-0124.

Introduction

Banyan networks [2] comprise a large class of networks that have been used for interconnection in large-scale multiprocessors and telephone switching systems. A Banyan network is a network in which there is a unique path from each input to each output.[1] Regular variants of Banyan networks, such as delta and butterfly networks, have been used in multiprocessors such as the IBM RP3 [6] and the BBN Butterfly [7]. Analysis of the performance of Banyan networks has typically focused on these regular variants.

Patel [5] presented a probabilistic analysis of the performance of delta networks. His work assumed that all sources transmit with uniform probability, and that all destinations are selected with uniform probability. Bhuyan [1] has extended Patel's work to include analysis of the case where each processor has a single favorite destination that is not the favorite destination of any other processor. Kruskal and Snir [3] have extended Patel's work by finding an asymptotic expression for the probability that a destination is receiving for networks with large numbers of stages.

In what follows, we present a methodology for performance analysis of general unbuffered Banyan networks. The analysis allows us to compute exactly the probability of successful message transmission in a Banyan network of arbitrary topology, under several assumptions:

1. The destination addresses for messages are unformly distributed over the outputs of the network.

2. The messages presented at each input are independent of the messages presented at other inputs, and also of messages presented on any previous cycle.

3. The network is fully synchronous, with all messages not dropped at stage n proceeding simultaneously to stage $n+1$ at each clock cycle.

Our methodology has two novel features: it does not assume that all sources transmit with equal probability and thus allows analysis of networks where some inputs are more likely to be active than others; and it allows analysis of Banyan networks of arbitrary topology.

Our work proceeds from the observation that all of the differing topologies for unqueued Banyan networks can be decomposed into combinations

[1] Or from each *base* to each *apex*, in the terminology of Goke and Lipovsky.

of three basic operations: bundling, concentration, and switching. By describing the behavior of these primitive elements with a probabilistic model, we are able to evaluate the performance of any such network.

We begin with a discussion of the use of probability mass functions to describe network wiring, and then consider the effect of each of three basic operations on these probability mass functions. Finally, we apply this model in an analysis of two switching elements, the common $2^k \times 2^k$ crossbar and the Transit RN1 switching element.

Modeling Message Traffic With Probability Mass Functions

A multi-stage network consists of a set of message sources, a cascaded set of network switching elements, and a set of message destinations. Often the set of source and destination nodes is identical.

The elements comprising a switching network are interconnected with channels. Each channel consists of a wire or group of wires that are switched as a single unit. A channel might, for example, consist of a single bidirectional wire with serial encoding of messages, or a byte-wide data path with an associated parity bit.

We associate with a channel a random variable l whose value is the number of messages, or the load, that the channel is carrying. The probability mass function (PMF) of this random variable specifies for each non-negative integer j the probability that the channel is carrying j messages. We call this function the *loading probability mass function* (LPMF) for the channel. If the random variable specifying the load for a channel is called l, then we denote the LPMF for the channel $p_l(l_0)$.

For example, a single channel has a probability p of carrying a message and a probability of $1-p$ of being idle. Thus the LPMF for a single channel is simply the PMF of a Bernoulli trial.

Our event space is the space of loading configurations for a particular network. That is, if we define a network as a set of message-carrying wires connected to each other by the switching elements we shall define below, then the elementary events in our event space are instances of this network with some load specified for each channel in the network. Obviously the $N+1$ possible values of l for a channel that can carry a maximum of N messages partition the event space into $N+1$ mutually exclusive sets of elementary events – each set containing all the network loading configurations for which the load on that channel is some given value.

In later sections, we will find it useful to associate with a probability

mass function its unilateral $\mathcal{Z}$-transform. We denote the $\mathcal{Z}$-transform of a PMF $p_x(x_0)$ by $p_x^T(z)$.

Bundling

The first operation is the simplest. We call the grouping together of several channels to form a single wider channel *bundling*. The single wider channel that is a product of bundling we sometimes call a *bundle*. The loads on the constituent channels in a bundle must be independent, as they will be in a Banyan network with independent inputs. When we bundle two channels, one of which can carry between 0 and n messages and the other of which can carry between 0 and m messages, the resulting channel can carry between 0 and $n + m$ messages. The loads of the channels being bundled are independent random variables whose sum we are forming, so that the LPMF of the resulting bundle will be the convolution of the LPMFs of the component channels. If we denote the bundling of a and b as $\mathcal{B}[p_a(a_0), p_b(b_0)]$, we have simply

$$\mathcal{B}[p_a(a_0), p_b(b_0)] \equiv p_a(a_0) * p_b(b_0)$$

where $*$ denotes convolution. In the $\mathcal{Z}$-domain, then, bundling will only require forming the product of the $\mathcal{Z}$-transforms:

$$\mathcal{Z}[\mathcal{B}[p_a(a_0), p_b(b_0)]] = p_a^T(z) \cdot p_b^T(z)$$

Figure 1 depicts the result of bundling eight channels, each of which carries a message with probability 1/2. The LPMF is clearly that of a binomial distribution, because the sum of independent identically distributed Bernoulli random variables is a binomial random variable.

Concentration

Our second elementary operation on channels is called *concentration*. In concentration, we take a bundle of M single channels and form from it a bundle of N single channels. If $N < M$, and the input bundle is carrying more than N messages, some messages will be lost.

The effect on the LPMF of the input bundle is simple. If $N \geq M$, there is no effect on the LPMF. If $N < M$, the probability that more than N messages can be carried on the output bundle is 0, but in cases where messages are dropped, only enough will be dropped to bring the load to N.

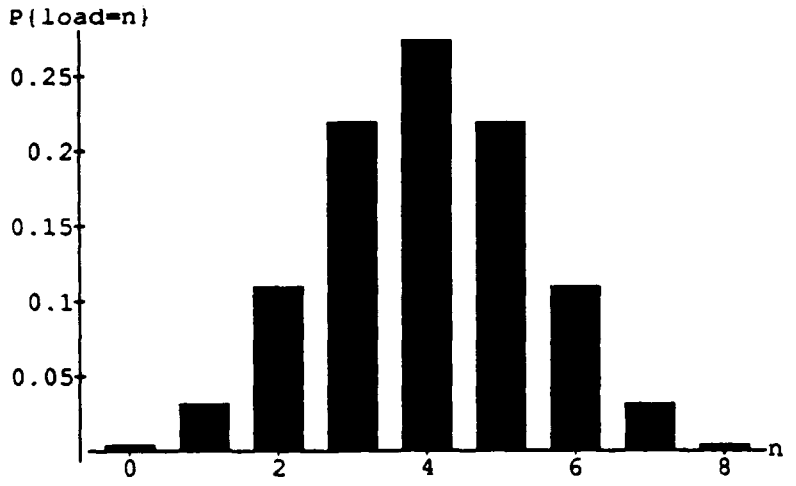

Figure 1: Loading probability mass function for an eight-channel bundle, where each channel carries a message with probability 1/2.

Thus the effect of the operation on an LPMF $p_l(l_0)$ will be both to clip it to 0 for $l_0 > N$ and to add to $p_l(N)$ the sum of $p_l(l_0)$ for $l_0 > N$. Figure 2 shows the result of concentration on the LPMF of figure 1.

More explicitly, if the input LPMF is given by

$$p_l(l_0) = \sum_{i=0}^{M} k_i \delta(l_0 - i)$$

where $\delta(n)$ is the unit impulse function, the result of N-concentration of $p_l(l_0)$, a bundle composed of M channels, to N channels, is given by

$$C_{M,N}\left[p_l(l_0)\right] \equiv p_l(l_0)\, u(N - l_0) + \left(\sum_{l_1=N+1}^{M} p_l(l_1) \right) \delta(l_0 - N)$$

where $u(n)$ is the unit step function.

If the $\mathcal{Z}$-transform of $p_l(l_0)$ is $p_l^T(z)$, then we have

$$\mathcal{Z}\left[C_{M,N}\left[p_l(l_0)\right]\right] = p_l^T(z) - \sum_{l_1=N+1}^{M} p_l(l_1)\, z^{l_1} + \left(\sum_{l_1=N+1}^{M} p_l(l_1) \right) z^N$$

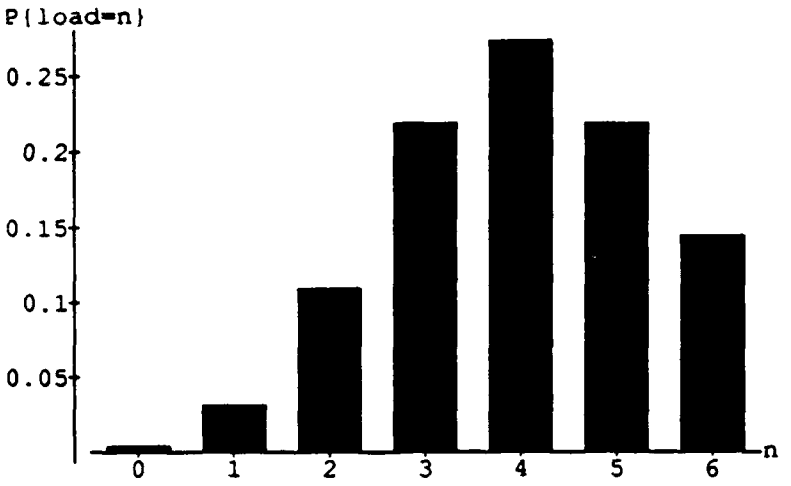

Figure 2: 6-concentration of the LPMF of figure 1.

The first two terms in the transform are the result of taking the $\mathcal{Z}$-transform of the truncated LPMF, and the last term adds in the $\mathcal{Z}$-transform of the increased final element of the LPMF. Combining the last two terms, we have

$$\mathcal{Z}\left[C_{M,N}\left[p_l(l_0)\right]\right] = p_l^T(z) + \sum_{l_1=N+1}^{M} p_l(l_1)\left(z^N - z^{l_1}\right)$$

Switching

The last operation we shall be using is switching. The switching operation is performed on an input bundle of N channels and specifies the LPMFs for two output bundles of N channels each. We designate the output bundles bundle 0 and bundle 1. We specify for the modeled switch the probability $(1-q)$ that the each message in the input bundle is switched to to channel 0; similarly, messages are switched to channel 1 with probability q.

We now consider the LPMFs for the two output bundles, given the LPMF $p_l(l_0)$ of the input bundle. Suppose the input bundle is carrying i messages. What is the probability that j messages, where $j \leq i$, will be switched to channel 1? It is the probability of j successes in i Bernoulli trials. If we call the random variable specifying the load on the output bundle l_s, we have

for the conditional probability

$$p_{l_s|l}(j|i) = \binom{i}{j} q^j (1-q)^{i-j}$$

We may now apply the theorem of total probability to find an expression for $p_{l_s}(j)$:

$$\begin{aligned} p_{l_s}(j) &= \sum_{i=0}^{N} p_l(i) p_{l_s|l}(j|i) \\ &= \sum_{i=j}^{N} p_l(i) \binom{i}{j} q^j (1-q)^{i-j} \end{aligned}$$

where the summation's lower bound is changed in the second expression because the probability that the output channel carries more messages than the input channel is 0.

Thus if S specifies the probability of switching a message to a given output bundle, and the input LPMF is given by $p_l(l_0)$, then the LPMF for the output bundle is given by

$$S\left[p_l(l_0), S\right] \equiv p_{l_s}(l_{s_0}) = \sum_{i=l_{s_0}}^{N} p_l(i) \binom{i}{l_{s_0}} S^{l_{s_0}} (1-S)^{i-l_{s_0}}$$

This can be interpreted as meaning that the probability that l_{s_0} messages appear on an output bundle is the probability that l_{s_0} messages were on the input bundle and all l_{s_0} messages were switched to the given output bundle, plus the sum of the probability that $l_{s_0}+1$ messages were on the input bundle and exactly l_{s_0} of these were switched to the given output bundle, and so on, up to the maximum possible load of the input bundle.

An example of the effect of switching may be seen in figure 3.

To evaluate the $\mathcal{Z}$-transform of $S\left[p_l(l_0), S\right]$, we first note that the random variable describing the number of messages on an output bundle may be treated as the sum of a random number of identically distributed random variables. We can see this by imagining individually switching each channel in the input bundle to one output bundle or the other, before considering whether it is carrying a message.

Then there is one random variable for each channel in the input bundle; call it b. b is 1 if the channel is switched to the output bundle being considered, and 0 if the channel is switched to the other output bundle. b's PMF

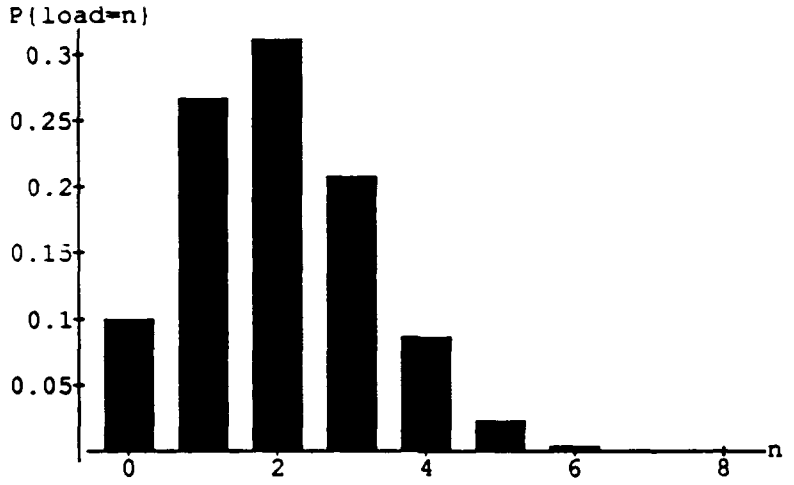

Figure 3: The effect of switching the LPMF of figure 1 with probability 0.5.

is then given by

$$p_b(b_0) = (1 - S)\delta(b_0) + S\delta(b_0 - 1)$$

for S the probability of switching to the given output bundle.

The random number of summands is the number of messages that the input bundle was actually carrying; thus its distribution is the LPMF of the input bundle. To extend our earlier interpretation, we are saying here that the load on a particular output bundle is the number of occupied channels in the input bundle that were switched to that output bundle.

Now the $\mathcal{Z}$-transform of an output channel's LPMF is given by the transform of the sum of a random number of identically distributed random variables:

$$\begin{aligned} \mathcal{Z}\left[\mathcal{S}\left[p_l(l_0), S\right]\right] &= p_l^T\left(p_b^T(z)\right) \\ &= p_l^T(1 - S + Sz) \end{aligned}$$

We note that, where the probability of switching to both bundles is equal, the $\mathcal{Z}$-transform for the LPMF resulting from repeated stages of switching has a particularly simple form. If b is the random variable representing the

number of channels switched to an output bundle, we have

$$p_b^T(z) = \mathcal{Z}\left[\frac{\delta(b_0) + \delta(b_0 - 1)}{2}\right] = \frac{z+1}{2}$$

We can now follow the rule given above, so that n levels of switching cause repeated substitutions for z, and we have the recurrence relation

$$\begin{aligned} S(n) &= \frac{S(n-1)+1}{2} \\ S(0) &= z \end{aligned}$$

with solution

$$S(n) = \frac{z + 2^n - 1}{2^n}$$

Thus, if l_i is the random variable for the load on the input bundle and l_n is the random variable for the load on an output bundle after n levels of binary switching with uniform probability of switching to either channel, we have

$$p_{l_n}^T(z) = p_{l_i}^T\left(\frac{z + 2^n - 1}{2^n}\right)$$

Descriptions of Simple Switching Elements

We describe two simple switching elements, the $2^k \times 2^k$ crossbar and the Transit RN1 switching element, by using combinations of our three primitive operations: bundling, concentration, and switching. In depicting the primitive operations schematically, we use the symbols shown in figure 4.

The $2^k \times 2^k$ Crossbar

The common $2^k \times 2^k$ crossbar network is formed by bundling the 2^k inputs, switching k times (once per bit of routing data), and concentrating the outputs with an 2^k-input, one-output concentrator. We depict the probabilistic model of an eight-by-eight crossbar in figure 5.

For a $2^k \times 2^k$ crossbar, we may find an output channel's LPMF as follows. If we call the probability that an input channel is transmitting Q_i, the LPMF for an input channel is given by

$$p_y(y_0) = Q_i\delta(y_0 - 1) + (1 - Q_i)\,\delta(y_0)$$

with $\mathcal{Z}$-transform

$$p_y^T(z) = Q_i z + (1 - Q_i)$$

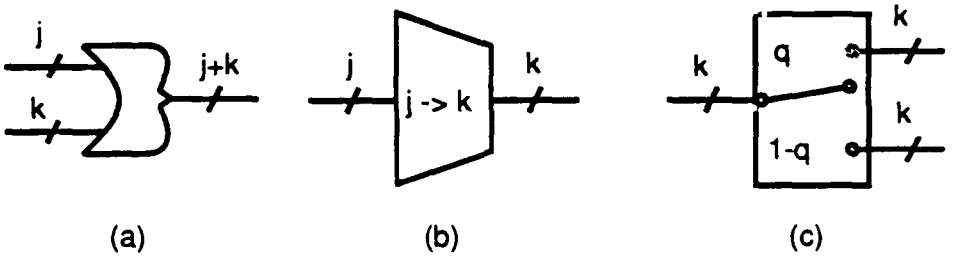

Figure 4: (a) The symbol for bundling two input bundles into one. (b) The symbol for concentrating j channels to k channels. (c) The symbol for switching with probability q to the top output channel, and $1 - q$ to the bottom output channel.

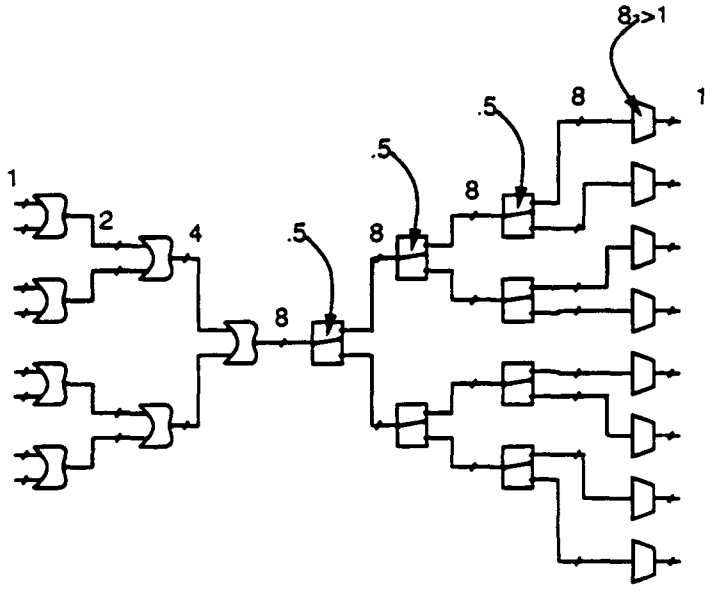

Figure 5: An eight-by-eight crossbar network.

The LPMF for a bundle of 2^k identical input channels has $\mathcal{Z}$-transform

$$p_{x_c}^T(z) = \left(p_y^T(z)\right)^{2^k}$$

The result of k stages of switching with equal probability in each of two directions is

$$\begin{aligned} p_{x_s}^T(z) &= \left(p_y^T\left(\frac{z+2^k-1}{2^k}\right)\right)^{2^k} \\ &= \left(Q_i\left(\frac{z+2^k-1}{2^k}\right)+(1-Q_i)\right)^{2^k} \\ &= \left(\frac{Q_i}{2^k}z+\left(1-\frac{Q_i}{2^k}\right)\right)^{2^k} \\ &= \sum_{l=0}^{2^k}\binom{2^k}{l}\left(1-\frac{Q_i}{2^k}\right)^l\left(\frac{Q_i}{2^k}z\right)^{2^k-l} \end{aligned}$$

We note that this $\mathcal{Z}$-transform is trivially invertible, so that, setting $M = 2^k$ and rearranging slightly, we have

$$\begin{aligned} p_{x_s}^T(z) &= \left(\frac{Q_i}{M}\right)^M\left(\sum_{l=0}^{M}\binom{M}{l}\left(\frac{M}{Q_i}-1\right)^l z^{M-l}\right) \\ p_{x_s}(x_{s_0}) &= \left(\frac{Q_i}{M}\right)^M\left(\sum_{l=0}^{M}\binom{M}{l}\left(\frac{M}{Q_i}-1\right)^l \delta(x_{s_0}-(M-l))\right) \end{aligned}$$

Now we may perform the $M \to 1$ concentration. Because this is a concentration to one channel, we may save some work by noting that we can simply consider the loading probability for zero messages from the LPMF above; the concentration forces all other loading probabilities to that for one message, which will necessarily be the complement of the loading probability for zero messages. We have, for the loading probability for zero messages,

$$p_{x_s}(0) = \left(\frac{Q_i}{M}\right)^M\left(\sum_{l=0}^{M}\binom{M}{l}\left(\frac{M}{Q_i}-1\right)^l \delta(-(M-l))\right)$$

Note that the terms in the summation are nonzero only where $l = M$, so this expression simplifies to

$$p_{x_i}(0) = \left(\frac{Q_i}{M}\right)^M \left(\frac{M}{Q_i} - 1\right)^M$$
$$= \left(1 - \frac{Q_i}{M}\right)^M$$

The LPMF of an output channel is then given by

$$p_l(l_0) = \left(1 - \frac{Q_i}{M}\right)^M \delta(l_0) + \left(1 - \left(1 - \frac{Q_i}{M}\right)^M\right) \delta(l_0 - 1)$$

As the number of stages k in the crossbar increases, $M = 2^k$ becomes large quickly, and $p_l(1)$ quickly approaches the limit

$$\lim_{k \to \infty} p_l(1) = \lim_{M \to \infty} \left(1 - \left(1 - \frac{Q_i}{M}\right)^M\right) = \left(1 - e^{-Q_i}\right)$$

In our analysis, the probability of successful message transmission is given by the ratio of the expected number of messages transmitted by all the output channels to the expected number of messages on input channels. In the case of a square crossbar network, this is simply $p_l(1)/Q_i$. We plot this value as a function of Q_i, the input loading on the network, for an eight-by-eight crossbar network in figure 6.

The Transit RN1 Switching Element

The RN1 switching element is a prototype for the switching element to be used in the Transit interconnection network for massively parallel computers, being built by the Transit Group at MIT's Artificial Intelligence Laboratory. The RN1 switching element can be configured in one of two ways; the first is as two four-by-four crossbars, and the second is as an eight-by-four crossbar with a dilation of two, meaning that only four logical output directions are available, but two messages can be carried in each. It is the second of these configurations whose performance we analyze. We depict the element in figure 7.

The derivation of the LPMF of a two-channel output bundle for the RN1 switching element follows. As above, if we call the probability that an input channel is transmitting Q_i, the LPMF for an input channel is given by

$$p_y(y_0) = Q_i \delta(y_0 - 1) + (1 - Q_i)\, \delta(y_0)$$

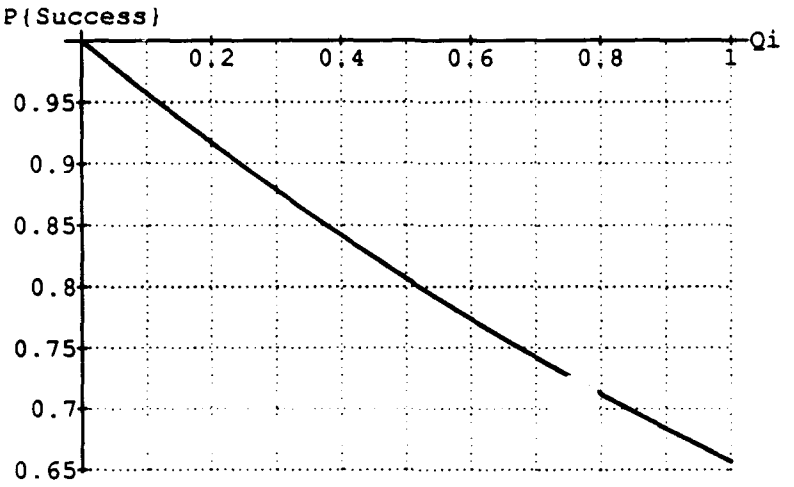

Figure 6: The probability of successful message transmission as a function of Q_i, the input loading on the network, for an eight-by-eight crossbar network.

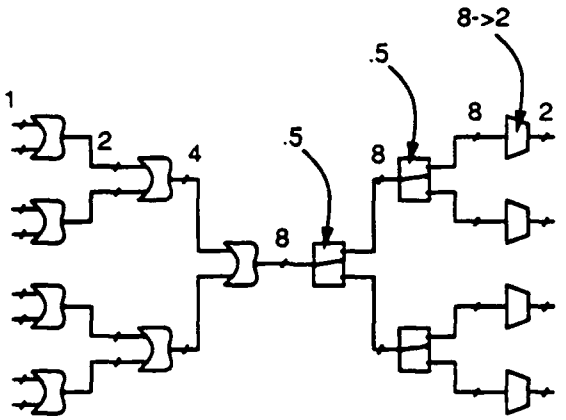

Figure 7: The RN1 switching element, in the eight-by-four, dilation two configuration.

with $\mathcal{Z}$-transform

$$p_y^T(z) = Q_i z + (1 - Q_i)$$

The LPMF for a bundle of 8 identical input channels has $\mathcal{Z}$-transform

$$p_{x_c}^T(z) = \left(p_y^T(z)\right)^8$$

The result of two stages of switching with equal probability in each of two directions is

$$\begin{aligned} p_{x_s}^T(z) &= \left(p_y^T\left(\frac{z+3}{4}\right)\right)^8 \\ &= \left(Q_i\left(\frac{z+3}{4}\right) + (1 - Q_i)\right)^8 \\ &= \left(\frac{Q_i}{4}z + \left(1 - \frac{Q_i}{4}\right)\right)^8 \\ &= \sum_{l=0}^{8}\binom{8}{l}\left(1 - \frac{Q_i}{4}\right)^l\left(\frac{Q_i}{4}z\right)^{8-l} \end{aligned}$$

Again as in the case of the crossbar, we invert the transform

$$p_{x_s}(x_{s_0}) = \left(\frac{Q_i}{4}\right)^8\left(\sum_{l=0}^{8}\binom{8}{l}\left(\frac{4}{Q_i} - 1\right)^l\delta(x_{s_0} - (8 - l))\right)$$

and then perform the concentration. In this case the concentration is to two channels, so that we must consider probabilities for the two cases that the output bundle carries zero messages or one message in order to use the method we did before for deriving the concentrated LPMF.

For zero messages, the sum is zero except where $l = 8$, so that we have:

$$\begin{aligned} p_{x_s}(0) &= \left(\frac{Q_i}{4}\right)^8\left(\frac{4}{Q_i} - 1\right)^8 \\ &= \left(1 - \frac{Q_i}{4}\right)^8 \end{aligned}$$

For one message, the sum is zero except where $l = 7$, so we have:

$$\begin{aligned} p_{x_s}(0) &= \left(\frac{Q_i}{4}\right)^8(8)\left(\frac{4}{Q_i} - 1\right)^7 \\ &= 2Q_i\left(1 - \frac{Q_i}{4}\right)^7 \end{aligned}$$

After concentration, the probability for two messages must be the sum of the probabilities for higher loads, so that the LPMF for a two-channel output bundle is given by

$$p_l(l_0) = \left(1 - \frac{Q_i}{4}\right)^8 \delta(l_0) + 2Q_i \left(1 - \frac{Q_i}{4}\right)^7 \delta(l_0 - 1) + \left(1 - \left(1 - \frac{Q_i}{4}\right)^7 \left(1 + \frac{7Q_i}{4}\right)\right) \delta(l_0 - 2)$$

We form the probability of successful message transmission as the ratio of the expectation of the number of messages on all output channels to the expectation of the number of messages on all input channels. In this analysis we have assumed uniformity and independence of input loading and a uniform distribution of message destinations, so that the expectation of the input loading is simply $\sum_{n=1}^{8} Q_i = 8Q_i$ and that of the output loading (if we recall that the random variable giving the number of messages on an output bundle is l) is

$$\mathrm{E}[4l] = 4\left(1 \cdot \left(2Q_i \left(1 - \frac{Q_i}{4}\right)^7\right) + 2 \cdot \left(1 - \left(1 - \frac{Q_i}{4}\right)^7 \left(1 + \frac{7Q_i}{4}\right)\right)\right)$$

Thus the probability of successful message transmission is given by

$$\begin{aligned} P_{SMT} &= \frac{Q_i \left(1 - \frac{Q_i}{4}\right)^7 + 1 - \left(1 - \frac{Q_i}{4}\right)^7 \left(1 + \frac{7Q_i}{4}\right)}{Q_i} \\ &= \frac{1 + \left(Q_i - \left(1 + \frac{7Q_i}{4}\right)\right) \left(1 - \frac{Q_i}{4}\right)^7}{Q_i} \\ &= \frac{1 - \left(1 + \frac{3Q_i}{4}\right) \left(1 - \frac{Q_i}{4}\right)^7}{Q_i} \end{aligned}$$

We plot the probability of successful message transmission versus the input loading in figure 8.

Analyzing the Performance of More Complex Networks

It may be difficult to simplify the expressions describing more complex networks built from arrays of simple switching elements like those we have analyzed above. Indeed, Patel [5] and Kruskal and Snir [3] derive expressions only for simple, regular networks; these are $a^n \times b^n$ delta networks in

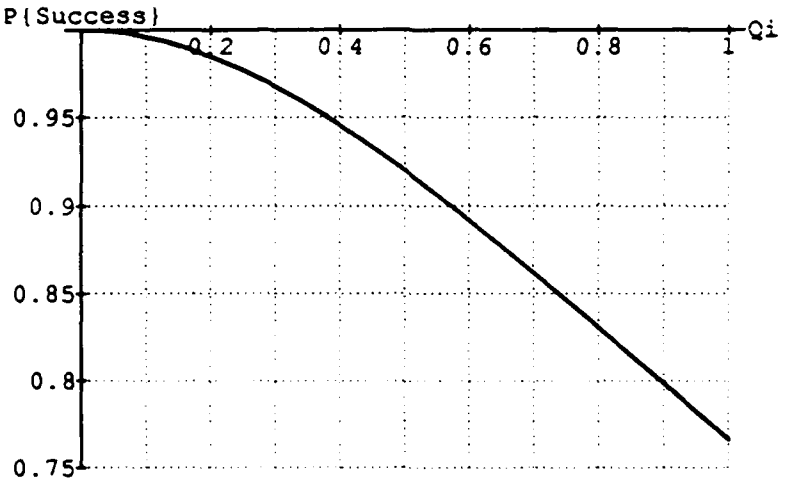

Figure 8: Probability of successful message transmission plotted vs. input loading for the Transit RN1 switching element in its eight-by-four, dilation two configuration.

the case of Patel's work, and square Banyan networks of arbitrary dilation in the case of Kruskal and Snir's work.

The advantage of our approach is that such analyses are automated. One specifies the topology of the network, forms the sequence of operators that describes the loading probability mass function for an output bundle, and evaluates it. This evaluation can be numeric or in the form of a parameterized expression. The derivation of such an expression for a complex network is aided by the use of a symbolic mathematics program like Macsyma or Mathematica. We present a set of Mathematica functions that may be used for such analysis in the appendix.

Future Work

The methodology described above, despite its advantages, does not yield a completely satisfactory model of a Banyan network's performance. We describe now some of the disadvantages of the methodology.

The probability of successful message transmission alone will not be a faithful measure of performance in a network that is buffered. In the Transit network, for example, although the individual switching elements themselves

do not contain buffers, messages are effectively buffered at the inputs to the network. Thus it will be desirable to extend the methodology with queueing-theoretic techniques to allow the creation of more faithful models. In future work, we hope to do this in a fashion that continues to allow analyses of Banyan networks of arbitrary topology.

While the methodology does allow analysis of networks where one or more sources are likely to be more active than others, it does not easily lend itself to an analysis of networks where one or more sinks are more likely to be the destination of messages than others. In fact, the general case of this problem, where messages entering a Banyan network of arbitrary topology have an arbitrary distribution of destination addresses, remains unsolved to date.

The analysis technique described is appropriate only to Banyan networks. While these constitute a large class, some of the fault-tolerance features of Banyan networks used in practice may include redundant paths between sources and sinks. It will be necessary to extend the technique to encompass replications and dilations of Banyan networks; in more complicated cases, it may be necessary to supplement it with a different approach, or abandon it altogether.

Another disadvantage of the methodology we have described lies in its tacit assumption that the network modeled is completely synchronous. This assumption is not always justified; for example, in the case of a circuit-switched network like the Transit network, successful message transmission creates a circuit which is held open until a reply is sent. The circuit is held for a number of cycles, during which other messages may be transmitted from the inputs and be blocked because paths at succeeding stages are already in use.

A related disadvantage of our methodology lies in its assumption that the path being built by a message being transmitted in a circuit-switched network immediately disappears, freeing all associated resources, if the message is blocked, whereas in reality it will take a number of cycles for these resources to be freed. Nussbaum and his colleagues have found this to be a significant factor in discrepancies between Patel's model and their simulation, as described in in [4].

We hope to address some of these disadvantages in ongoing work. The ideal result would be a technique for automatically generating an accurate model of the performance of a multistage interconnection network given only a description of the network topology.

Appendix: Mathematica Functions for Banyan Network Analysis

```
concentrate::usage =
    "concentrate[x, n] concentrates the LPMF x to n channels."

concentrate[x_, n_] :=
    (* get distribution for 0 through n-1 channels, and add
       as last element the sum of the rest of the channels. *)
    Append[Take[x, n], Apply[Plus, Drop[x, n]]]

discreteconvolution::usage =
    "discreteconvolution[x, y] treats x and y as 0-based
     vectors and returns their discrete convolution."

discreteconvolution[x_, y_] :=
    Block[{xlgth, ylgth, lgth},
          xlgth = Length[x];
          ylgth = Length[y];
          lgth = xlgth + ylgth - 1;
          (* in summation, portions of sequence with indices
             out of range for sequences must be treated as
             0. *)
          Table[Sum[If[k < 1 || k > xlgth ||
                       (n-k+1) < 1 || (n-k+1) > ylgth,
                       0,
                       (* because of the 0->1 index
                          translation, we increase the y-index
                          to shift the result sequence back
                          down to begin at 1. *)
                       x[[k]] y[[n-k+1]]],
                    {k, xlgth}],
                {n, lgth}]]

bundle::usage =
    "bundle[x, y] forms the LPMF that results from bundling
     two input bundles with LPMFs x and y."

bundle[x_, y_] :=
    discreteconvolution[x, y]
```

```
switch::usage =
    "switch[x, p] returns the LPMF of an output bundle to
     which x is switched with probability p."

switch[x_, p_] :=
    Block[{lgth},
          lgth = Length[x];
          Table[Sum[x[[i+1]] Binomial[i, n] p^n (1-p)^(i-n),
                    {i, n, lgth-1}],
                {n, 0, lgth-1}]]
```

Bibliography

References

[1] Bhuyan, L. N. "An Analysis of Processor-Memory Interconnection Networks," in *IEEE Transactions on Computers*, Vol. C-34, No. 3, March 1985.

[2] Goke, L. R., and Lipovski, G. J. "Banyan Networks for Partitioning Multiprocessor Systems," in *Proceedings of the First Annual Symposium on Computer Architecture*, 1973.

[3] Kruskal, C. P., and Snir, Marc. "The Performance of Multistage Interconnection Networks for Multiprocessors," in *IEEE Transactions on Computers*, Vol. C-32, No. 12, December 1983.

[4] Nussbaum, D., Vuong-Adlerberg, I., and Agarwal, A. "Modeling a Circuit-Switched Multiprocessor Interconnect," in *Proceedings of of the 1990 ACM SIGMetrics Conference on Measurement and Modeling of Computer Systems*, May, 1990.

[5] Patel, J. H. "Performance of Processor-Memory Interconnections for Multiprocessors," in *IEEE Transactions on Computers*, Vol. C-30, No. 10, October 1981.

[6] Pfister, G. F. *et al.*, "The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture," in *Proceedings of the 1985 International Conference on Parallel Processing*, August, 1985.

[7] Rettberg, R., and Thomas, R. "Contention is no Obstacle to Shared-Memory Multiprocessing," in *Communications of the ACM*, Vol. 29, No. 12, December, 1986.